



Repeat and Missing Number Array

Link submit: <https://www.interviewbit.com/problems/repeat-and-missing-number-array/>

Solution:

C++	http://ideone.com/ymDdna
Java	https://ideone.com/uyW2od
Python	https://ideone.com/exXJte

Tóm tắt đề: Cho một dãy số A gồm N số nguyên dương, các số nguyên dương có giá trị nằm trong tập $[1 \dots N]$. Các số nguyên dương này khác nhau, duy nhất chỉ có 2 số có giá trị giống nhau. Bạn hãy tìm ra giá trị của số giống nhau và giá trị của số không xuất hiện trong dãy.

Kết quả: Bạn trả về một vector `<int>`, vector này sẽ chứa 2 phần tử, phần tử thứ 1 là giá trị của phần tử giống nhau, phần tử thứ 2 là giá trị của phần tử không xuất hiện trong dãy.

[3 1 2 5 3]	[3 4]
-------------	-------

Giải thích: Trong dãy đã cho, có 2 phần tử có giá trị là 3 giống nhau, và không có phần tử mang giá trị là 4 xuất hiện trong dãy.

Hướng dẫn giải:

Hướng tiếp cận của bài toán:

Ta nhìn thấy rằng hàm của chúng ta khi truyền vào là một vector hằng, tức là ta không được phép thay đổi dữ liệu của vector đó.

- Hướng giải 1: Ta sử dụng một mảng `cnt[]`, với ý nghĩa `cnt[x]` là số lượng phần tử `x` xuất hiện trong mảng A. Như vậy, ta duyệt một vòng lặp từ 0 đến $N - 1$, với mỗi phần tử `i`, ta tăng `cnt[A[i]]` lên 1 đơn vị. Sau đó, ta lại duyệt các phần tử `x` từ 1 đến N, nếu như `cnt[x] = 2`, ta kết luận `x` là phần tử bị lặp lại 2 lần. Nếu `cnt[x] = 0`, ta kết luận `x` là phần tử bị mất trong dãy. Đánh giá độ phức tạp: $O(N)$ về mặt thời gian, $O(N)$ về mặt không gian.

Tuy nhiên, bài toán này lại còn có một cái khó khăn nữa, đó là bạn chỉ được phép sử dụng bộ nhớ với độ phức tạp $O(1)$ mà thôi. Tức là bạn phải sử dụng một số lượng biến là hữu hạn, không thể sử dụng mảng phụ thuộc vào kích thước N như thế. Như vậy ta làm thế nào?

Bạn để ý: Đề cho bạn giá trị các số trong dãy A chỉ thuộc đoạn $[1 \dots N]$. Thứ 2 là các phần tử trong dãy A lại có giá trị khác nhau, chỉ có duy nhất 2 thằng trùng nhau thôi! Điều này có nghĩa rằng dãy số chỉ sai khác dãy số từ 1 đến N đúng 2 phần tử.

Như vậy, bạn gọi a là giá trị của phần tử bị trùng lặp, gọi b là giá trị của phần tử bị mất.

Bạn gọi $\text{tongbac1} = 1 + 2 + 3 + \dots + N = N * (N - 1) / 2$

Bạn gọi $\text{tongbac2} = 1^2 + 2^2 + 3^2 + \dots + N^2 = N * (2*N + 1) * (N + 1) / 6$

Bạn gọi $\text{sum1} = A[1] + A[2] + A[3] + \dots + A[N]$

Bạn gọi $\text{sum2} = A[1]^2 + A[2]^2 + A[3]^2 + \dots + A[N]^2$

Như vậy rõ ràng, bạn sẽ giải được hệ phương trình như sau:

$$a - b = \text{sum1} - \text{tongbac1}$$

$$a^2 - b^2 = \text{sum2} - \text{tongbac2}$$

Tới đây, bạn hoàn toàn có thể giải được bài toán giải một hệ phương trình 2 ẩn khi biết 2 phương trình đã cho.

Độ phức tạp thời gian: $O(N)$, độ phức tạp không gian: $O(1)$.